

symbian
OS

A guide to Symbian Signed



A guide to Symbian Signed

3rd Edition, 03/08

Published by:

Symbian Software Limited

2-6 Boundary Row

Southwark

London SE1 8HP

UK

www.symbian.com

Trademarks, copyright, disclaimer

'Symbian', 'Symbian OS' and other associated Symbian marks are all trademarks of Symbian Software Ltd. Symbian acknowledges the trademark rights of all third parties referred to in this material. © Copyright Symbian Software Ltd 2008. All rights reserved. No part of this material may be reproduced without the express written permission of Symbian Software Ltd. Symbian Software Ltd makes no warranty or guarantee about the suitability or the accuracy of the information contained in this document. The information contained in this document is for general information purposes only and should not be used or relied upon for any other purpose whatsoever.

Compiled by:

Ben Morris

Managing Editor:

Ashlee Godwin

Design Consultant:

Sabeena Aslam

Reviewed by:

Roderick Burns

Bruce Carney

Ricky Junday

Khalid Mohammed

WHAT’S NEW IN SYMBIAN SIGNED? 6

SIGNING IN CONTEXT 6

OPTIONS FOR SIGNING.....7

BENEFITS OF OWNING A PUBLISHER ID.....7

OBTAINING A PUBLISHER ID.....8

SYMBIAN OS CAPABILITIES.....9

HOW TO USE SYMBIAN OS CAPABILITIES.....10

IMEI-BASED RESTRICTIONS.....10

TESTING IN CONTEXT.....10

WHY TEST?.....10

INTRODUCTION TO THE SYMBIAN SIGNED WEBSITE.....11

OPEN SIGNED — ONLINE.....11

HOW TO SIGN APPLICATIONS USING OPEN SIGNED — ONLINE WITHOUT A PUBLISHER ID 12

Step 1: Go to the Symbian Signed public website and access the service.....12

Step 2: Confirm your email address.....12

Step 3: Access your email account to download your signed application 12

RESTRICTIONS AND LIMITS.....12

CHECKLIST 13

OPEN SIGNED— OFFLINE.....13

HOW TO SIGN APPLICATIONS USING OPEN SIGNED—OFFLINE WITH A PUBLISHER ID 13

Step 1: Register for a Symbian Signed account 13

Step 2: Download the Developer Certificate Request creation tool.....13

Step 3: Run the tool to generate a Developer Certificate Request CSR file 14

Step 4: Upload the Developer Certificate Request file to the Symbian Signed portal.....15

Step 5: Sign your applications with your Developer Certificate 16

RESTRICTIONS AND LIMITS.....17

CHECKLIST 17

EXPRESS SIGNED 17

HOW TO SIGN APPLICATIONS USING EXPRESS SIGNED 17

Step 1: Register for a Symbian Signed account 18

<i>Step 2: Purchase Content IDs.....</i>	<i>18</i>
<i>Step 3: Sign your application using SignSIS and your Publisher ID</i>	<i>19</i>
<i>Step 4: Submit your Publisher ID signed application SIS file to the portal</i>	<i>19</i>
<i>Step 5: Download your Symbian Signed application from the portal.....</i>	<i>21</i>
<i>Step 6: Auditing and Test Criteria Compliance</i>	<i>22</i>
RESTRICTIONS AND LIMITS	23
CHECKLIST.....	24
CERTIFIED SIGNED.....	24
HOW TO SIGN APPLICATIONS USING CERTIFIED SIGNED	24
<i>Step 1: Register for a Symbian Signed account.....</i>	<i>24</i>
<i>Step 2: Sign your application using SignSIS and your Publisher ID.....</i>	<i>25</i>
<i>Step 3: Submit your Publisher ID signed application SIS file to your chosen Test House.....</i>	<i>25</i>
<i>Step 4: Download your Symbian Signed application from the portal.....</i>	<i>26</i>
RESTRICTIONS AND LIMITS	27
CHECKLIST.....	28
SYMBIAN SIGNED TEST CRITERIA.....	28
TIPS FOR TESTING	29
GET THE BASICS RIGHT.....	29
UNDERSTAND THE TEST CASES	29
MAKE CONTINUOUS TESTING PART OF YOUR METHODOLOGY	30
UIDS FOR SYMBIAN DEVELOPMENT	31
FREQUENTLY ASKED QUESTIONS	34
SYMBIAN SIGNED WITH MANUFACTURER CAPABILITIES FOR NOKIA.....	36
SYMBIAN SIGNED WITH MANUFACTURER CAPABILITIES FOR SONY ERICSSON	37
WHO'S WHO IN THE SYMBIAN SIGNED PROCESS?.....	37
SUPPORT FOR SYMBIAN SIGNED.....	39
GLOSSARY.....	40

What's New in Symbian Signed?

Symbian Signed has changed recently, introducing new and simplified signing options for applications, and a new lower cost Certificate Authority (CA). No matter what kind of application you are developing for Symbian OS, whether it is commercial or non-commercial, the changes should make it easier for you to get your software signed and deployed.

The following three signing options are now available:

- *Open Signed*, Developer Certificate based signing, including a completely new online-only signing option for developers without a Publisher ID.
- *Express Signed*, a streamlined signing option that does not require independent testing.
- *Certified Signed*, the mainstream signing option based on independent testing by a Symbian-accredited Test House.

The number of Capabilities requiring Device Manufacturer approval has been minimized, and a simpler, unified process has been created for applications that do still require manufacturer approval.

Independent testing is now only required for *Certified Signed*. However, all applications are still expected to satisfy any test cases relevant to them. The Symbian Signed Test Criteria, which have been revised and updated, can be found on the Symbian Signed website at www.symbiansigned.com.

Signing in Context

Signing is the process of encoding a tamper-proof digital certificate into an application. The certificate identifies the application's origin, and grants access to those Capability-protected APIs in Symbian OS that the application declared at build-time.

On Symbian OS, protected APIs are those that allow sensitive operations, such as those that may:

- access end users' private data, thus potentially breaching privacy
- potentially create billable events, thus costing the end user money
- access the mobile phone network, potentially affecting its operation
- access handset functions that can affect the normal behavior of the phone
- potentially impact the performance of other applications running on the phone.

Developers creating straightforward applications should find it possible to avoid the signing process altogether by not using Capability-protected APIs. Alternatively, where certain Capabilities are required, the developer may rely on the user to grant blanket permission to the application at install time, or 'single shot' permission at run time (for example, agreeing to send a message) if the security policy of the Device Manufacturer allows it. Although this is possible on some devices at the time of writing, there is no guarantee that Device Manufacturers or network operators will always allow unsigned applications to install on their devices.

There is no requirement to sign applications targeted at versions of Symbian OS earlier than v9.x. However, developers working with pre-v9 releases should consider the merits of migrating their applications to the latest versions of Symbian OS.

Options for Signing

Symbian Signed gives developers different options for getting their applications signed.

- *Open Signed* makes it easy for developers to sign applications for limited deployment to known devices, either for testing or for personal use. Deployment is restricted by device IMEI.
- *Express Signed* offers a fast and cost effective signing route for most applications, with some restrictions on the Capabilities available. There is no requirement for independent testing prior to signing for those developers who own a Publisher ID and are releasing commercial software. Developers without a Publisher ID (e.g., freeware and shareware) can also feasibly access this signing option via publisher distribution channels.
- *Certified Signed* provides access to all but Device Manufacturer Capabilities, but requires applications to be submitted for independent testing. *Certified Signed* applications are entitled to use the 'for Symbian OS' logo to aid differentiation and brand building.



The costs of the different options vary depending on whether a Publisher ID is required, and on whether independent testing is required. Open Signed is free as neither is required when signing online.

The prerequisites for the different signing options are as follows:

	Publisher ID Required	Independent Testing Required	IMEI Restrictions	For Commercial Distribution?
Open Signed Online	NO	NO	YES	NO
Open Signed Offline	YES	NO	YES	NO
Express Signed	YES	NO	NO	YES
Certified Signed	YES	YES	NO	YES

Benefits of Owning a Publisher ID

Publisher ID digital certificates form part of the Public Key Infrastructure, and are issued by Certificate Authorities. The Certificate Authority for Symbian Signed is TC TrustCenter (but existing ACS Publisher IDs issued by VeriSign remain valid for some signing options). Developers can purchase Publisher IDs directly from TC TrustCenter at www.trustcenter.de/order/publisherid/dev.

Symbian Signed provides several signing options for developers who do not own a Publisher ID:

- *Open Signed* enables developers to sign and deploy applications on a limited scale for testing, and for non-commercial and personal use, without requiring a Publisher ID.

- Publisher Channel partners offer signing options for developers who are unable to acquire a Publisher ID.

However, as the cost of acquiring a Publisher ID is relatively low, developers may want to consider the benefits of purchasing their own:

- Owning a Publisher ID allows developers to request and use Developer Certificates with much greater flexibility than is possible without a Publisher ID, and enables much larger scale deployment, allowing larger scale beta testing, for example.
- Owning a Publisher ID allows access to more signing options, and gives developers control over publisher identity and branding.
- Trust is important to the end users of your application and for ultimately building a positive experience for all mobile applications. Owning a Publisher ID allows you to enhance your reputation for delivering trusted applications.

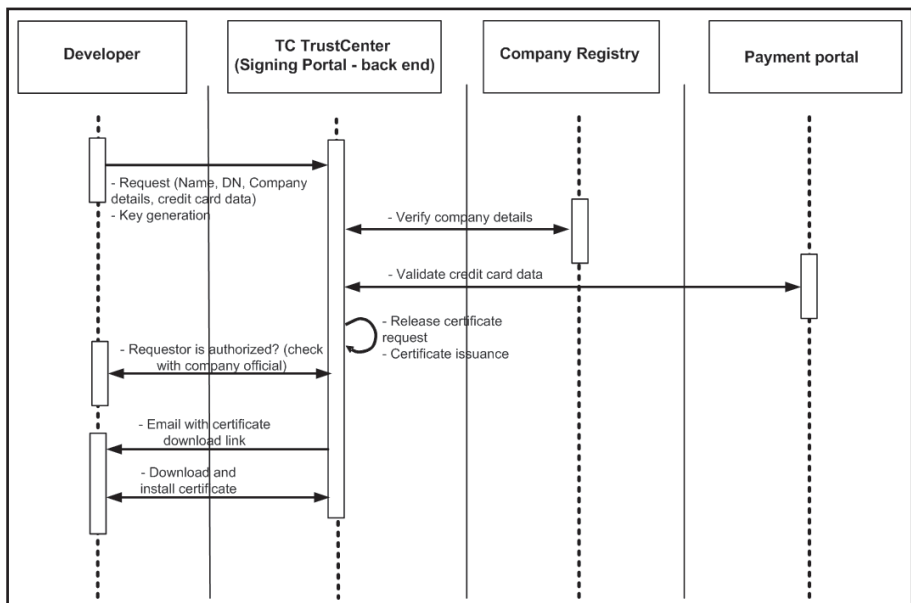
Obtaining a Publisher ID

Publisher IDs can be purchased from TC TrustCenter using the following link:
www.trustcenter.de/order/publisherid/dev.

Ensure that you use Internet Explorer to apply for the Publisher ID. You must also use the same PC and internet browser for both applying for and downloading the Publisher ID.

There are some steps that must be followed so that the identity of you and the company you work for can be verified. Once this is complete the Publisher ID will be issued by TC TrustCenter.

The following diagram shows a high level view of this process:



The applicant will have to provide personal identification and the company will have to provide documents proving its existence.

Once you have downloaded the Publisher ID it is necessary to extract the Certificate and Key files. This is done using the `tcp12p8` tool provided at the following link:
developer.symbian.com/wiki/display/sign/Symbian+Signed+Tools.

Once you have downloaded the tool you should copy the tool files and the PFX or P12 file forming your Publisher ID into the same location. Now open a command line interface for this location and type:

```
Tcp12p8.bat <name of the Publisher Id .pfx file> <password for the pfx file> yourkeyfile.key yourcerfile.cer
```

When you come to sign a SIS file or create a DevCert request you will require the CER file, KEY file and the password for your Publisher ID.

Symbian OS Capabilities

Symbian Signed enables applications to use the Platform Security architecture and distinguishes between User System and Restricted Capabilities. The various signing options allow applications to request different Capabilities:

- User Capabilities are available through all signing options.
- All System Capabilities, including Restricted (as defined in the table below), are available through *Open Signed* (with a Publisher ID) and *Certified Signed* options.
- *Express Signed* does not allow access to Restricted Capabilities (CommDD, DiskAdmin, NetworkControl, and MultimediaDD).
- Symbian Signed refers to the most sensitive Capabilities, specifically AllFiles, DRM, and TCB, as Device Manufacturer Capabilities. These are only available through the *Open Signed* (with a Publisher ID) and *Certified Signed* options and require Device Manufacturer approval.

This is summarized in the following table:

Capability Type	Capability Name	Description	Availability
User Capabilities	LocalServices Location NetworkServices ReadUserData UserEnvironment WriteUserData	• User Capabilities are designed to be meaningful to mobile phone users • Depending on Device Manufacturer security policies, users may be able to grant blanket or single-shot permission to applications which use these Capabilities	All signing options
System Capabilities	PowerMgmt ProtServ ReadDeviceData SurroundingsDD SwEvent TrustedUI WriteDeviceData	• System Capabilities that protect system services, device settings, and some hardware features	All signing options
Restricted Capabilities	CommDD DiskAdmin NetworkControl MultimediaDD	• Restricted Capabilities that protect file system, communications, and multimedia device services	<i>Open Signed</i> (with Publisher ID) and <i>Certified Signed</i> options only
Device Manufacturer Capabilities	AllFiles DRM TCB	• Trusted Computing Base and System Capabilities that protect the most sensitive system services	Require Device Manufacturer approval

How to Use Symbian OS Capabilities

The most complete guide to using Capabilities is the Symbian Press book *Symbian OS Platform Security* (Craig Heath, 2006). More details about the book can be found on the Symbian Developer Network website, developer.symbian.com/books, under **Symbian Press › Developer titles**.

In addition, the Symbian OS Library documentation, available in SDKs and online from the Symbian Developer Network website, provides a complete and up-to-date list of which APIs are Capability-protected.

IMEI-Based Restrictions

Open Signed is specifically intended to limit deployment to devices for testing or for personal use; in consequence, applications signed using the *Open Signed* option are restricted by IMEI to specific devices.

Device IMEIs are declared as part of the signing process.

	Number of devices allowed
Publisher ID plus Device Manufacturer approval	> 1000
Publisher ID	1000
No Publisher ID	1

Testing in Context

Testing is an essential part of all software development. Symbian Signed defines specific tests to ensure a minimum level of robustness and stability for applications running on Symbian OS phones.

Tests are defined in the Symbian Signed Test Criteria, and are divided into two main groups:

- Universal Tests (prefixed UNI), which test for basic application reliability and robustness, including: stress testing; correct basic behavior, such as correct installation, uninstall and reinstall; and compliance with system event, Task List requirements and scalable UI.
- Capability Related Tests (prefixed CAP), which test against specific Symbian OS v9.x features, including Platform Security-related behavior and Internet phone features. All applications are expected to comply with the Universal Tests, and applications that use certain Capabilities are required to comply with the Capability Related Tests.

Further details on tests can be found later in this booklet, in the Symbian Signed Test Criteria section.

Why Test?

In a consumer-focused market such as mobile phones, quality is a critical success factor for applications, even more so than features and functionality. You should design robust and effective testing procedures into every step of your development activity.

Writing robust, reliable, efficient, and secure native applications for mobile devices that meet end user expectations is challenging. The Symbian Signed Test Criteria is designed to help developers ensure that all third-party applications written for Symbian OS achieve a minimum level of quality.

To do so, it defines a minimum set of test cases that all applications should pass. Although tests alone cannot guarantee correct application behavior, they do provide confidence that applications meet specific conditions, for example:

- applications must not block incoming calls, overwrite file systems, or refuse to shut down
- in resource-critical scenarios (such as low memory) applications must be well-behaved
- when uninstalled, applications must not leave any installation files behind (so, for example, they cannot bury malicious code somewhere on the end user's phone).

Application stability is important not just to end users, but to Device Manufacturers, mobile network operators and other application developers too. Testing is therefore an essential foundation for the Symbian Signed service.

Introduction to the Symbian Signed website

The Symbian Signed website comprises two main sections:

Symbian Signed Overview and **My Symbian Signed**.

The **Symbian Signed Overview** section contains links to information about the Test Houses, Documentation and the Symbian Signed Test Criteria. All of this information is stored on the Symbian Developer Network website, developer.symbian.com.

The **My Symbian Signed** section allows a user to submit and manage applications, to purchase Content IDs and to use the *Open Signed* option of Symbian Signed.

The Symbian Signed wiki on the Symbian Developer Network (developer.symbian.com/wiki/display/pub/Symbian+Signed) contains important information and tools for the Symbian Signed process.

Open Signed — Online

Open Signed makes it easy for developers to sign applications for limited deployment to known devices, either for testing or for personal use. *Open Signed* applications are signed against a Developer Certificate, and application deployment is restricted by device IMEI.

Using the *Open Signed Online* signing option, no Publisher ID is required. Developer Certificate signing is performed via the portal, on behalf of the developer, providing a rapid, free option for one-off signing of applications for use on a single device, restricted by IMEI.

In many circumstances *Open Signed* meets the needs of freeware, open source, and personal use developers, as well as of those experimenting with Symbian OS or the signing process. It is also useful to developers working on unsupported host development platforms (for example, Linux or Mac OS X) who have problems running the tools required for other signing options.

The main features of this signing option are:

- the process is online
- no Publisher ID is required
- no Symbian Signed account is required

- no tool downloads are required, so it is platform independent
- all User and System Capabilities may be requested (excludes the Restricted Capabilities and Device Manufacturer Capabilities)
- applications are restricted to one device, specified by IMEI
- there is no cost to developers.
- if UIDs from the protected range are used, they must have been allocated to the account associated with the email address
- UIDs from the "development range" may be used.

How to Sign Applications Using Open Signed — Online Without a Publisher ID

This signing option requires no Symbian Signed account, no tools download, and no Publisher ID. You will need to supply a valid email address as part of the submission process. Signed applications will be stored by the portal for 30 days from completion of signing.

Step 1: Go to the Symbian Signed public website and access the service

The information you provide will be encoded into a Developer Certificate which the portal will generate and use to sign your application. You will not be able to download the Developer Certificate.

1. Go to www.symbiansigned.com.
2. From the **Welcome** page click on the **Open Signed Online** link to go to the online submission page.
3. Enter the requested information in the online form, including device IMEI and your email address.
4. Select the Capabilities required by your application.
5. On your local machine, navigate to the SIS file of the application you are submitting for signing.
6. Enter the security code displayed and prompted for.
7. Click to view the legal agreement.
8. Click on **Accept** after you have read the legal agreement.

Step 2: Confirm your email address

An email will be sent to you containing a link to confirm your email address. Click the link to confirm.

Step 3: Access your email account to download your signed application

An email will be sent to you containing a link to your Developer Certificate-signed application SIS file. Click the link to download your application.

Restrictions and Limits

- This signing option is not to be used for any form of commercial distribution of applications.
- If UIDs from the protected range are used, they must have been allocated to the account associated with the email address.
- UIDs outside the protected range may be used.
- No Publisher ID is required so there is no validation of developer identity, except for a confirmation that the email address supplied is live at the time of certificate issue. Therefore, end users may be presented with an install time warning and prompted to complete installation.
- Applications are restricted by device IMEI to one device.
- No access to Restricted Capabilities or Device Manufacturer Capabilities.
- Applications are signed with a Developer Certificate against the Symbian A Root certificate.
- Signed applications are valid for 36 months from the date of signing.

Checklist

- ☐ Requires a valid email account.
- ☐ Developers must list the Capabilities required by the application as part of the online process, and accept an online legal agreement.
- ☐ Device IMEI must be known.

Open Signed— Offline

Open Signed makes it easy for developers to sign applications for limited deployment to known devices for testing and development. *Open Signed* applications are signed against a Developer Certificate, and application deployment is restricted by device IMEI. Using this signing option, developers who own a Publisher ID can request Developer Certificates allowing them to sign applications for deployment on up to 1000 devices, with access to all required User and System Capabilities.

The main features of this signing option are:

- a Publisher ID is required
- a Symbian Signed account is required
- all User, System and Restricted Capabilities may be requested
- applications are restricted by IMEI, but up to 1000 devices may be specified
- the request process generates a Developer Certificate which is downloaded and used for local signing of applications
- Developer Certificates can be used to sign multiple SIS files.

How to Sign Applications Using Open Signed—Offline with a Publisher ID

To use this signing option, you must have a Symbian Signed account. Your **My Symbian Signed** account page provides access to all tools and processes needed to sign your applications and manage and track the online signing processes. You will also need a Publisher ID issued by TC TrustCenter, at www.trustcenter.de/order/publisherid/dev.

Existing ACS Publisher IDs supplied by VeriSign remain valid for this signing option.

As part of this process you will be required to run Symbian's DevCertRequest tool to generate a certificate request CSR file.

Step 1: Register for a Symbian Signed account

If you do not already have one, you will need to register for a Symbian Signed account at www.symbiansigned.com.

You will need to supply a valid email address, to which an account activation link will be sent.

Access your email account, and click the link to activate your new Symbian Signed account. The email address you supply will become your Symbian Signed user name; your password is the one you provide when you register initially.

Step 2: Download the Developer Certificate Request creation tool

To obtain a Developer Certificate, you must first generate a certificate request CSR file; you submit the CSR file to the Symbian Signed portal, which generates the Developer Certificate based on the information contained in the CSR. You can then download and use the certificate to sign your applications.

You can find this tool on the Symbian Developer Network using the following link:
developer.symbian.com/wiki/display/pub/Symbian+Signed+Tools

The page includes a download link for the DevCertRequest tool, `DevCertRequest.exe`, a self-installing Microsoft Windows application wizard.

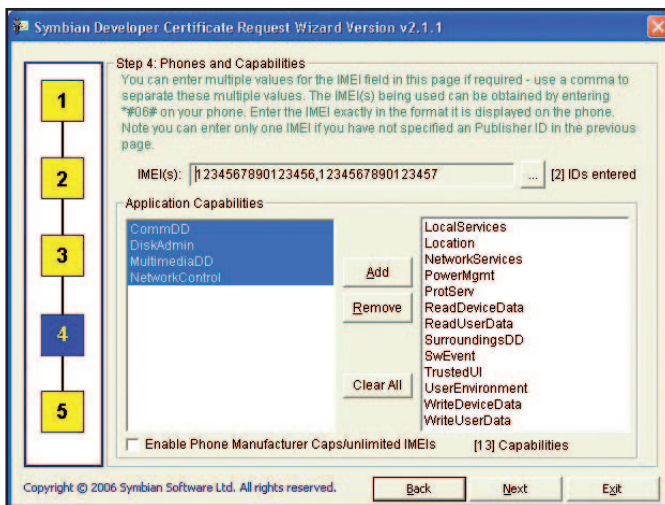
Step 3: Run the tool to generate a Developer Certificate Request CSR file

To generate a CSR certificate request file you will need access to your Publisher ID certificate file and the associated private key and password.

1. Run `DevCertRequest.exe` on your local machine to generate a CSR certificate request file (a CSR is a standards-conforming encrypted file which the portal uses to generate the signed Developer Certificate).
2. When you run the tool, a wizard launches. You will be asked to:
 - supply a name for the CSR file to be generated
 - supply a Publisher ID and the associated private key and password
 - enter the identification information which is to appear in the Developer Certificate
 - specify the IMEI(s) of the device(s) to which your application will be deployed
 - specify the Capabilities your application requires. If you need access to Device Manufacturer Capabilities or wish to request a Developer Certificate for more than 1000 IMEIs, select the **Enable Manufacturer Caps** button
 - the information you provide is encoded into the CSR file and used to generate the Developer Certificate, for example:

```
Certificate Request File: W:\ADevCertRequest.CSR
Private Key File: C:\DOCUME~1\ADeveloper\adeveloper.private.key
Country: UNITED KINGDOM
State: N/A
City: London
Company: My Company Limited
Common Name: A Developer
IMEI(s): XXyyXyXy-NnnNnN-1
App Capabilities: PowerMgmt, ReadDeviceData, WriteDeviceData,
TrustedUI, ProtServ, SwEvent, NetworkServices, LocalServices,
ReadUserData, WriteUserData, Location, SurroundingsDD,
UserEnvironment
```

3. The wizard offers you an option to view the contents of the CSR file. Confirm that the information you have supplied is correct. The wizard will generate a CSR file.



Step 4: Upload the Developer Certificate Request file to the Symbian Signed portal

1. Log in and go to your **My Symbian Signed** account page at www.symbiansigned.com.
2. From the left navigation bar, click on the **Open Signed > Request** link to go to the certificate request page.
3. Enter the security code displayed and prompted for.
4. Browse on your local machine to the CSR file generated at Step 3 and click the **Send** button.

Address <https://www.symbiansigned.com/app/barclayhtml/devcert/requestupload.jsp>

Google Go

Symbian Signed Overview **My Symbian Signed**

Tools
Submissions
TCT Content IDs
UIIDs
Open Signed
 Request
 My DevCerts
Legal
Documentation and support

Symbian.com

McAfee
OPERATED BY
CIDERCONE

New Open Signed Developer Certificate request

Please be aware if you are requesting a developer certificate for SonyEricsson phone(s) that makes use of one of the capabilities below then a [capability request form](#) MUST be completed and send to devcerts@sonyericsson.com before the request is made.

- AllFiles
- TCB
- DRM

Please notice that Capability request form needs to be filled in and sent to SonyEricsson only once. After the requested capabilities have been granted once, you don't need to request approval anymore for the same purpose. However, if you need some other capabilities or use the granted capabilities for some other purpose than what they were originally granted for, please send the updated form to SonyEricsson for the new approval.

Please ensure you have used the Developer Certificate Request tool to create your CSR file. If you have not used this tool then your uploaded file will not be valid and a developer certificate will not be issued!

Generating a DevCert can take a few minutes...please wait while your request is being processed.

Please type in the security code seen in the picture below using only **letters A-F** and **numbers 0-9**

Please attach the PKCS#10 formatted certificate request file:

5. Wait for the file to be uploaded; you will be notified when the upload is complete.
6. The portal will generate a Developer Certificate file based on the information you provided in your CSR certificate request. All certificates which you generate are listed with their expiry dates in the **Existing Certificates** list. To locate the newly created Developer Certificate, from your **My Symbian Signed** account page, go to the left navigation bar and click on the **Open Signed > MyDevCerts** link.
7. View the certificate contents using the **Show** button, and verify that the specified IMEI(s) and the Capabilities granted are correct.

The screenshot shows the 'My Symbian Signed' account page. On the left is a navigation menu with links: Tools, Submissions, TCT Content IDs, UIDs, Open Signed (selected), Request, My DevCerts, Legal, Documentation and support, and Symbian.com. Below the menu is a McAfee logo and the text 'OPERATED BY CIDERCOVE'. The main content area is titled 'View DevCert' and includes the text 'This applies to Symbian OS v9 only'. Below this is a section 'Existing Certificates' with a table listing certificates. Each row has buttons for 'Show' and 'Download'. At the bottom of the table are links for '[Prev]' and '[Next]'.

ID	Date requested	Exp. date	Capabilities / IMEIs	Show	Download
2261817	2008-02-28	2011-02-27		Show	Download
2261812	2008-02-28	2011-02-27		Show	Download
2260646	2008-02-21	2011-02-20		Show	Download
2260322	2008-02-20	2011-02-19		Show	Download
2247464	2008-02-15	2011-02-14		Show	Download
2228819	2008-02-11	2011-02-10		Show	Download
1889412	2008-01-10	2011-01-09		Show	Download
1889388	2008-01-10	2011-01-09		Show	Download
1853990	2008-01-07	2011-01-06		Show	Download
1853989	2008-01-07	2011-01-06		Show	Download
1612537	2007-12-13	2010-12-12		Show	Download
1612535	2007-12-13	2010-12-12		Show	Download
1612534	2007-12-13	2010-12-12		Show	Download
1612533	2007-12-13	2010-12-12		Show	Download
1612532	2007-12-13	2010-12-12		Show	Download

[Prev] [Next]

8. Download and store the Developer Certificate on your local machine, and ensure that you also store the password which is associated with it and which you provided to `DevCertRequest.exe` (it is a good idea to reserve a dedicated directory for storing all certificates and keys). You can now use your Developer Certificate to sign your application SIS files.

If you need Device Manufacturer Capabilities (`AllFiles`, `TCB` and `DRM`), you will need to follow a different process after you upload your CSR file:

1. From the left navigation bar, click on the **Open Signed > Request** link to go to the certificate request page.
2. Click on the **Request Phone Manufacturer Approved DevCert** link at the bottom of the page.
3. Select the Device Manufacturer from the drop-down list and supply the details requested in the subsequent pages. The Device Manufacturer will be notified of the request after you submit your CSR file. Once the manufacturer approves your request, you can download the certificate from your **My Symbian Signed** account page as before.

See pages 36 and 37 for details of how to request Phone Manufacturer Capabilities

Step 5: Sign your applications with your Developer Certificate

Application signing is performed locally by the developer using the Developer Certificate CER certificate file, its associated password, and SignSIS or MakeSIS tools.

For detailed guides to signing and installing application SIS files, refer to the Software Installation Toolkit guide and reference, under **Tools and Utilities** in the Symbian OS Library documentation, available in SDKs and online from Symbian Developer Network, developer.symbian.com.

Restrictions and Limits

- This signing option is not intended for commercial distribution of applications.
- Applications are restricted by device IMEI, with an upper limit of 1000 devices.
- Access to Device Manufacturer Capabilities (`AllFiles`, `DRM` and `TCB`) requires manufacturer approval.
- A Developer Certificate is valid for 36 months from date of issue. Applications are valid for the remaining life of the certificate from the date of signing.
- A Developer Certificate may be used to sign an unlimited number of applications during its lifetime.

Checklist

- ☐ Valid email and Symbian Signed accounts are required.
- ☐ A valid Publisher ID is required.
- ☐ Developers must download the current version of the `DevCertRequest.exe` tool, which requires Microsoft Windows.
- ☐ The SignSIS or MakeSIS tools are required to sign a SIS file.
- ☐ Device IMEIs must be known.

Express Signed

Express Signed is intended for the general release of applications, including commercial and non-commercial applications. It is also suitable for the general release of enterprise applications by large organizations. It is available to any developer, organization or company that owns a Publisher ID.

The main features of this signing option are:

- a Publisher ID is required
- a Symbian Signed account is required
- User and System Capabilities may be requested (Restricted Capabilities cannot be requested)
- unrestricted application deployment
- signed applications are valid for ten years from the date of signing
- single applications and multiple batched applications may be submitted
- independent testing is not required, but applications must meet the Symbian Signed Test Criteria, and must have been tested on that basis before submission. Applications are audited for compliance and may be revoked.

For in-house developers, or commercial or professional developers with their own Publisher IDs, *Express Signed* provides a streamlined signing path without external dependencies.

How to Sign Applications Using Express Signed

To use this signing option, you must register for a Symbian Signed account. Your **My Symbian Signed** account page provides access to sign your applications, and to manage and track the online signing processes. You will also need a Publisher ID issued by TC TrustCenter at www.trustcenter.de/order/publisherid/dev, or you will need to work through an affiliate program which provides you with access to a Publisher ID.

Note that ACS Publisher IDs supplied by VeriSign cannot be used for *Express Signed* submissions, although existing ACS Publisher IDs remain valid for *Certified Signed*.

Step 1: Register for a Symbian Signed account

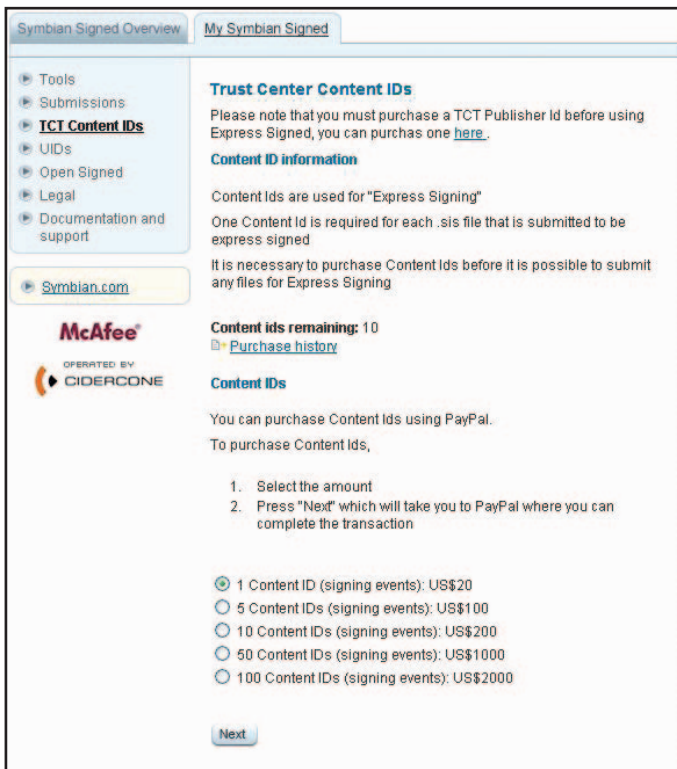
If you do not already have one, you will need to register for a Symbian Signed account at www.symbiansigned.com. You will need to supply a valid email address, to which an account activation link will be sent. (Please note that Symbian Signed only accepts registration from privately registered domains or company domains; public email domains and ISP domains are not accepted.)

Access your email account, click the link and follow the instructions to activate your new Symbian Signed account.

The email address you supply will become your Symbian Signed user name; your password is the one you provide when you register initially.

Step 2: Purchase Content IDs

Each submission for *Express Signing* involves a fee in the form of one pre-paid Content ID per application signed. Content IDs can be purchased through your Symbian Signed account using Paypal, and are stored within your account. For your submission to succeed, you will need to have sufficient Content IDs in your account. In the **My Symbian Signed** section of the site select the **TCT Content IDs** option and use Paypal to purchase Content IDs for *Express Signing*.



The screenshot shows the 'My Symbian Signed' page. On the left is a navigation menu with links: Tools, Submissions, **TCT Content IDs** (highlighted), UIDs, Open Signed, Legal, Documentation and support, and Symbian.com. Below the menu is a McAfee logo with the text 'OPERATED BY CIDERCOONE'. The main content area is titled 'Trust Center Content IDs' and contains the following text: 'Please note that you must purchase a TCT Publisher Id before using Express Signed, you can purchas one [here](#).' Below this is 'Content ID information' stating that Content IDs are used for 'Express Signing', one Content ID is required for each .sis file, and it is necessary to purchase Content IDs before submitting files. It shows 'Content ids remaining: 10' and a 'Purchase history' link. Under 'Content IDs', it says 'You can purchase Content Ids using PayPal.' and 'To purchase Content Ids,' followed by a numbered list: 1. Select the amount, 2. Press 'Next' which will take you to PayPal where you can complete the transaction. At the bottom, there is a list of purchase options with radio buttons: 1 Content ID (US\$20), 5 Content IDs (US\$100), 10 Content IDs (US\$200), 50 Content IDs (US\$1000), and 100 Content IDs (US\$2000). A 'Next' button is at the very bottom.

Symbian Signed Overview | **My Symbian Signed**

- Tools
- Submissions
- TCT Content IDs**
- UIDs
- Open Signed
- Legal
- Documentation and support
- Symbian.com

McAfee
OPERATED BY
CIDERCOONE

Trust Center Content IDs

Please note that you must purchase a TCT Publisher Id before using Express Signed, you can purchas one [here](#).

Content ID information

Content Ids are used for "Express Signing"

One Content Id is required for each .sis file that is submitted to be express signed

It is necessary to purchase Content Ids before it is possible to submit any files for Express Signing

Content ids remaining: 10
[Purchase history](#)

Content IDs

You can purchase Content Ids using PayPal.

To purchase Content Ids,

1. Select the amount
2. Press "Next" which will take you to PayPal where you can complete the transaction

- ☒ 1 Content ID (signing events): US\$20
- ☐ 5 Content IDs (signing events): US\$100
- ☐ 10 Content IDs (signing events): US\$200
- ☐ 50 Content IDs (signing events): US\$1000
- ☐ 100 Content IDs (signing events): US\$2000

[Next](#)

Step 3: Sign your application using SignSIS and your Publisher ID

To sign your application you must either have your own Publisher ID, or access to the Publisher ID of the organization for which or through which you are submitting the application. You must use the SignSIS or MakeSIS tool locally to sign the application SIS file.

For detailed guides to signing and installing application SIS files, refer to the Software Installation Toolkit guide and reference, under **Tools and Utilities** in the Symbian OS Library documentation, available in SDKs and online from the Symbian Developer Network, developer.symbian.com.

Step 4: Submit your Publisher ID signed application SIS file to the portal

Once you have signed your application SIS file with your own Publisher ID, or the Publisher ID of the organization which you are submitting through, you are ready to submit your application.

1. Zip your application SIS file together with the PKG file from which you created it, a `readme.txt` release notes file, and a PDF user manual or plain text How-to document, making sure you have specified no paths in the ZIP file.

The screenshot shows the 'Symbian Signed' web portal. The main heading is 'Application submission form: new'. Below this, it says 'Step 9: Upload test file' and includes a note: 'Please note that you can [cancel submission](#) at any time.' The form is at '1/1' and asks for the 'Application file name:'. A note states: 'Note that the files are zipped without preserving the file directory paths (if using WinZip do not select the 'Save full path info' check box).' There is a text input field for the 'Test file (only zip-archives are allowed)' with a 'Browse...' button. A warning message says: 'A test file may be required where a standalone DLL may need to be Symbian Signed and the only way it can be tested is by using a "tester" application which executes all the functionality in the DLL. Please upload such an application in a ZIP file if it is appropriate for your submission. Failure to do so now may result in delays to the testing.' There is a 'Comments for test file' text area. At the bottom, a 'Submit' button is circled in red. A link for 'Step 8: Declarative statement' is at the very bottom.

Symbian Signed

Symbian Signed Overview | **My Symbian Signed**

- Tools
- Submissions**
 - Express Signed
 - Certified Signed**
 - Submit freeware
 - My applications
- TCT Content IDs
- UIDs
- Open Signed
- Legal
- Documentation and support

[Symbian.com](#)

McAfee
OPERATED BY
CIDRACONE

Application submission form: new

Step 9: Upload test file

Please note that you can [cancel submission](#) at any time.

1/1

Application file name:

Note that the files are zipped without preserving the file directory paths (if using WinZip do not select the 'Save full path info' check box).

Test file (only zip-archives are allowed)

[Browse...](#)

A test file may be required where a standalone DLL may need to be Symbian Signed and the only way it can be tested is by using a "tester" application which executes all the functionality in the DLL. Please upload such an application in a ZIP file if it is appropriate for your submission. Failure to do so now may result in delays to the testing.

Comments for test file

[Submit](#)

[Step 8: Declarative statement](#)

2. Log in and go to your **My Symbian Signed** account page at www.symbiansigned.com.
3. From the left navigation bar, click on the **Submissions > Express Signed** link and follow the steps described on the submission page.
 - Supply the developer, organization, and contact information requested.
 - Select either an **Application** or **Passive content** submission as appropriate. An Application is a standard SIS file, while Passive content is a non-executable SIS file, for example, a stub SIS file or a theme. (Unless you know that you are submitting a stub SIS file, your SIS file is a standard application SIS file).
 - Supply the application information requested, including name, description, version number, targeted handsets, programming language used, and application language.
 - You can also choose to have your application included in the catalog of Symbian Signed applications that Symbian maintains and shares with ecosystem partners.
 - Supply the test result information requested. If you enter *Fail* against any test, your submission will fail. If you claim an exception on a test ensure that you select this option and provide an explanation of the exception.
 - You will also be asked to complete declarative statements for your application. For pre Symbian OS v9 these include:
 - o any network connections or short-link sessions it initiates
 - o any billable events it initiates
 - o whether it accesses and, if so, how it uses any personal or PIM data held on the handset
 - o a justification for requesting sensitive Capabilities.

For Symbian OS v9, you will be required to provide information about the Capabilities that your software is using.

These statements must be completed as fully as possible on submission. Failure to do so may result in your application being revoked and/or your Symbian Signed account being limited.

Q13) Does your application use the following capability:
WriteUserData

☐ No

☒ Yes, please provide your rationale for using this capability

If yes, please outline the APIs used in this capability.

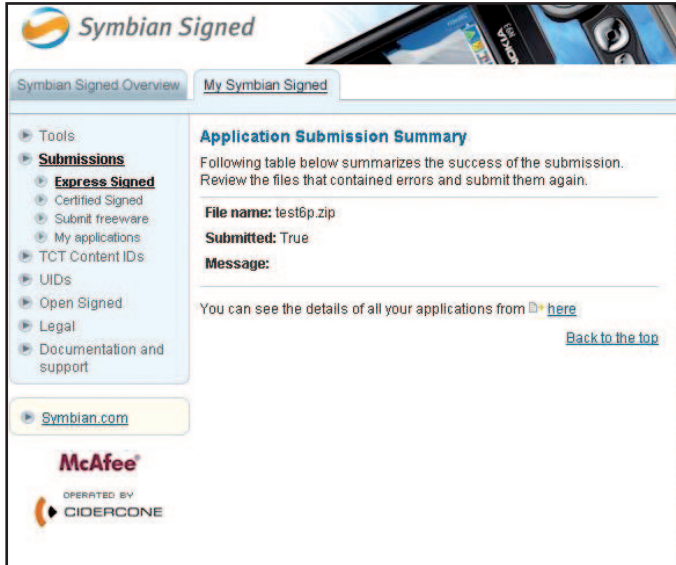
+ Step 6: Application test results

Submit

[Back to the top](#)

[legal disclaimer](#) | [Privacy policy](#) | [Symbian.com](#) | [Symbian Media Center](#) | [Support](#) | 2006 Symbian Ltd

4. On your PC, navigate to your submission ZIP file and click on the **Submit** link to complete your submission. When the file upload is complete, the portal scans and validates the files you submitted and notifies you whether the submission was accepted.



Step 5: Download your Symbian Signed application from the portal

Once your submission is accepted, your application SIS file is sent to the Certificate Authority for signing against the Symbian B Root certificate. Your Publisher ID will be validated and, if valid, the application will be signed. The signed application will be stored on the portal for you to download.

1. To download your signed application, log in and go to your **My Symbian Signed** account page at **www.symbiansigned.com**.
2. From the left navigation bar, click on the **Submissions > My Applications** link. Your Symbian Signed application will be listed next to a clickable link under the **Applications** heading.
3. Download your application by clicking on the link.

The screenshot shows the 'My Symbian Signed' page. On the left is a navigation menu with links: Tools, Submissions (selected), Express Signed, Certified Signed, Submit freeware, My applications (selected), TCT Content IDs, UIDs, Open Signed, Legal, and Documentation and support. Below the menu is the McAfee logo and the text 'OPERATED BY CIDERONE'. The main content area is titled 'Application status history' and 'Application details'. It displays the following information for application 'test6p':

- Id:** 33231
- Name:** test6p
- Date of submission:** Mar 13, 2008 4:32:57 PM EET
- Description:** test
- Additional accessories or add on hardware:**
- Testing round:** 1
- Terminals:** Nokia E61i
- Application type:** Entertainment
- Current status:** Accepted
- Declarative statement:** [View](#)
- SIS FileScan Results:** [View](#)
- Application in catalog:** no [Edit](#)
- Version number:** 1
- License:** Commercial
- Is the SIS file content?:** no
- Software languages:** C++
- Languages:** English
- Marketing information:**
- Marketing contact:** test
- Marketing email:** test
- Marketing phone:** test

At the bottom, there is a red text prompt: 'Please fill in the 3 question [feedback form](#) so we can continue to improve Symbian Signed and the service provided by the test houses.' Below this is a download link: '+ [Download signed application](#) test6p.zip (6.5 kB, Mar 13, 2008 4:33:52 PM EET) Express signed version.' The status at the very bottom is 'No test file'.

Step 6: Auditing and Test Criteria Compliance

All of your Symbian Signed applications remain permanently listed under the **Applications** heading in your **My Applications** page.

If an application you have signed via *Express Signed* is selected for random audit, you will see its status change from *Accepted* to *In Audit*.

Being selected for audit need not affect the decisions you make about your application release or distribution, and you may choose to continue with your release plans unchanged. You can download the Symbian Signed file while the audit is in progress. The outcome of the audit may impact your future projects as described below.

Audited applications undergo testing by an accredited Independent Test House against the Symbian Signed Test Criteria in a similar way to *Certified Signed* applications, except that the process is managed on your behalf by Symbian Signed, and that there are no associated costs.

Two outcomes are possible:

Your application passes the tests

1. When selected for auditing, your application status is updated to *In Testing*.
2. When testing is complete your application status is updated to *Accepted*, indicating that you have passed the Audit.
3. The Test House test report is stored in the same place as your Symbian Signed application and is available for you to download.

Your application fails the tests

1. When selected for auditing, your application status is updated to *In Testing*.
2. When testing is complete your application status is updated to *Rejected*.
3. You will be notified via email that your application has failed testing during an audit.
4. *Express Signing* will be automatically disabled on your account.
5. The Test House test report is stored in the same place as your Symbian Signed application and is available for you to download.

Before a decision is made to change an application to the *Rejected* state, a Review Board consisting of industry representatives decides on a case-by-case basis whether or not a test failure should be considered reasonable, for example, if you documented exceptions to specific tests which do not apply to your application.

If the Board's decision is that your application's test failure was due to poor design or poor implementation, then the following steps will be taken:

1. The failed application may be revoked.
2. *Express Signing* access will be blocked temporarily on your account.
3. Your next two Symbian Signed submissions will be required to be made via *Certified Signed*, to ensure that a full test cycle is performed and that your applications successfully pass independent testing by an accredited Independent Test House.
4. *Express Signing* will be re-enabled on your account.

The goal is to ensure that all Symbian Signed applications comply with the Symbian Signed Test Criteria and satisfy the baseline requirements for reliability and robustness.

Restrictions and Limits

- Only available to developers owning a valid Publisher ID, or having access to one through an organization.
- Only Publisher ID digital certificates obtained from TC TrustCenter can be used for *Express Signed* signing.
- Signing is not free. Each signing submission uses one Content ID. Developers must purchase Content IDs through the website.
- Waivers are not permitted for *Express Signed* submissions, approved or otherwise.
- No access to Restricted Capabilities (CommDD, DiskAdmin, NetworkControl and MultimediaDD).
- No access to Device Manufacturer Capabilities (AllFiles, DRM and TCB).
- Signed applications are valid for ten years from the date of signing.
- Applications are traceable to software publishers via Publisher IDs; Publisher IDs and SIS files may be revoked by Symbian Software Ltd.
- Symbian Signed carry out random auditing of submitted applications. The purpose is to verify that applications meet the Symbian Signed Test Criteria, and to validate the application against the submission declarations that were made for it.
- *Express Signed* applications are not entitled to display the 'for Symbian OS' logo.

Checklist

- ☐ Valid email and Symbian Signed accounts are required.
- ☐ A valid Publisher ID is required.
- ☐ The SignSIS tool is required to sign a SIS file.
- ☐ Developers should be familiar with the Symbian Signed Test Criteria.
- ☐ Applications are required to comply with the Symbian Signed Test Criteria.

Certified Signed

Certified Signed is the most complete signing option for commercial third-party applications.

Because it involves independent testing, it is considered the most trusted of all signing options, and therefore allows applications to request all Capabilities, as well as Device Manufacturer Capabilities through the appropriate channel. *Certified Signed* applications have the right to use the '*for Symbian OS*' logo.

The main features of this signing option are:

- A Publisher ID is required.
- A Symbian Signed account is required.
- Applications must be submitted to a Symbian Signed accredited Test House as part of the signing process.

- All User, System and Restricted Capabilities may be requested. Access to Device Manufacturer Capabilities requires manufacturer approval.
- Signed applications are valid for ten years from the date of signing.
- Single applications and multiple batched applications may be submitted.
- Unrestricted application deployment.
- A "Fast Track" option is available for *Certified Signed*.

How to Sign Applications Using Certified Signed

To use this signing option, you must have a Symbian Signed account. Your **My Symbian Signed** account page provides access to all tools and processes needed to sign your applications and to manage and track the online signing processes. You will also need a Publisher ID issued by TC TrustCenter at www.trustcenter.de/order/publisherid/dev, and you must submit your application for independent testing by a Symbian Signed accredited Test House.

Existing ACS Publisher IDs supplied by VeriSign will also be accepted.

Step 1: Register for a Symbian Signed account

If you do not already have one, you will need to register for a Symbian Signed account at www.symbiansigned.com. You will need to supply a valid email address, to which an account activation link will be sent. Access your email account, and click the link to activate your new Symbian Signed account.

The email address you supply will become your Symbian Signed user name; your password is the one you provide when you initially register.

Step 2: Sign your application using SignSIS and your Publisher ID

To sign your application you must have your own Publisher ID. You must use the SignSIS tool locally to sign the application SIS file.

For detailed guides to signing and installing application SIS files, refer to the Software Installation Toolkit guide and reference, under **Tools and Utilities** in the Symbian OS Library documentation, available in SDKs and online from Symbian Developer Network at developer.symbian.com.

Your application will be independently tested against the Symbian Signed Test Criteria; you should ensure that your application meets the criteria before submitting it for testing.

Step 3: Submit your Publisher ID signed application SIS file to your chosen Test House

Once you have signed your application SIS file with your Publisher ID, submit it for signing.

1. Zip your application SIS file together with the PKG file from which you created it, a `readme.txt` release notes file, and a PDF user manual or plain text How-to document, making sure you have specified no paths in the ZIP file.
2. Log in and go to your **My Symbian Signed** account page at www.symbiansigned.com.
3. From the left navigation bar, click on the **Submission > Certified Signed** link and follow the steps described on the submission page:
 - Supply the developer, organization, and contact information requested.
 - Supply the application information requested, including name, description, version number, targeted handsets, programming language used, and application language.
 - Select the application type from the drop down list, to indicate whether this is an application, a DLL only, a server, and so on.
 - Specify any additional hardware which will be required to test the application.
 - Select your Test House of choice, and accept the legal agreement.
 - You can also choose to have your application included in the catalog of Symbian Signed applications that Symbian maintains and shares with its ecosystem partners.
4. Browse to your submission ZIP file, and click on the **Submit** link to complete your submission. When the file upload is complete, the portal scans and validates the files which you submitted and notifies you whether the submission was accepted.

The screenshot shows the 'My Symbian Signed' page on the Symbian Signed portal. The left sidebar contains a navigation menu with 'Tools' and 'Submissions' (highlighted). Under 'Submissions', there are links for 'Express Signed', 'Certified Signed' (highlighted), 'Submit freeware', 'My applications', 'TCT Content IDs', 'UIDs', 'Open Signed', 'Legal', and 'Documentation and support'. Below the menu is a 'Symbian.com' link and a McAfee logo with the text 'OPERATED BY CIPHERSTONE'. The main content area is titled 'Application submission form: new' and 'Step 9: Upload test file'. It includes instructions about canceling submissions, the application file name 'test6p.zip', and a note about zipping files. There is a text input field for the test file name and a 'Browse...' button. Below this is a large text area for 'Comments for test file'. At the bottom of the form is a 'Submit' button and a link to 'Step 8: Declarative statement'.

Ensure that you enter all the necessary information for the Declarative Statements. Failure to enter this information will result in your submission being rejected.

Step 4: Download your Symbian Signed application from the portal

Once your submission is accepted, your application SIS file is sent to the Test House which you nominated for testing and signing. Your Publisher ID will be verified against the submitter details you provided to ensure they match. If your Publisher ID is valid, and if your application passes all relevant tests, the application will be signed. The signed application will be stored on the portal for you to download.

Each Test House operates its own fee scales. For current testing fees and information on the services the Test Houses provide, refer to the following page:

developer.symbian.com/wiki/display/sign/Symbian+Signed+Test+Houses.

You can track the progress of your submission from your **My Symbian Signed > My Applications** page. You will also be notified of the progress of your submission by the Test House.

1. Once your application is in the *Accepted* state you can download your signed application, log in and go to your **My Symbian Signed** account page at www.symbiansigned.com.
2. From the left navigation bar, click on the **Submissions > My Applications** link. Your Symbian Signed application will be listed next to a clickable link under the **Applications** heading.

3. Download your application by clicking on the link.

The screenshot shows the 'My Symbian Signed' interface. On the left is a navigation menu with links like Tools, Submissions, Express Signed, Certified Signed, Submit freeware, My applications, TCT Content IDs, UIDs, Open Signed, Legal, and Documentation and support. The main area is titled 'Application status history' and 'Application details'. It displays information for application 'test6p' (Id 33231), submitted on Mar 13, 2008. Details include description 'test', testing round 1, terminals 'Nokia E61i', application type 'Entertainment', and current status 'Accepted'. There are links to view the declarative statement and SIS FileScan Results. Below this, it shows 'Application in catalog' (no), 'Version number 1', 'License Commercial', 'Is the SIS file content?' (no), 'Software languages C++', 'Languages English', and 'Marketing information'. At the bottom, there is a 'Download signed application' link for 'test6p.zip' (6.5 kB, Mar 13, 2008 4:33:52 PM EET) and a 'feedback form' link.

Restrictions and Limits

- Only available to developers with a valid Publisher ID.
- Publisher ID digital certificates issued by TC TrustCenter and existing ACS Publisher IDs from VeriSign are valid for this signing option.
- Signing is not free. Independent testing is required by this option and will be charged for by the Test House.
- All Capabilities may be requested. Device Manufacturer Capabilities require manufacturer approval.
- Signed application certificates are valid for ten years from the date of signing.
- Applications are traceable to software publishers via Publisher IDs; Publisher IDs and SIS files may be revoked by Symbian Signed.
- *Certified Signed* applications are the only applications entitled to display the 'for Symbian OS' logo and branding.

Checklist

- ☐ Valid email and Symbian Signed accounts are required.
- ☐ A valid Publisher ID is required.
- ☐ The SignSIS tool is required to sign a SIS file.
- ☐ Applications must pass independently administered tests against the Symbian Signed Test Criteria. Developers should ensure that their applications meet the test criteria, to avoid unnecessary test failures.

Symbian Signed Test Criteria

The Symbian Signed Test Criteria are designed to ensure that all third-party applications written for Symbian OS achieve a minimum quality level. All third-party Symbian OS applications must comply with the test criteria, and successfully pass all relevant test cases.

This applies whether or not you intend to submit your application for independent testing. Note that since *Express Signed* applications are randomly audited, your application may be subject to Test House testing, even when using this signing option.

The Symbian Signed Test Criteria document is available from the Symbian Developer Network at the following link: developer.symbian.com/sstestcriteria.

Test cases do not attempt to cover subjective or content criteria, including usability, localization, or UI style guide compliance, and they do not provide device coverage testing. They test only on the device types specified in the PKG file.

The test cases fall into two categories:

- Universal Tests (prefixed UNI) which emphasize basic criteria for application reliability:
 - o Applications must install, uninstall, and reinstall correctly, leaving no installation files behind after uninstall, and must install correctly from external mass memory storage (for example, memory cards).
 - o Applications must back up and restore successfully, and use appropriate file creation locations.
 - o Applications must survive stress tests including low memory at start-up, handling exceptional events like OOM (Out Of Memory) and power down or rebooting while running, and must handle service interruptions, as well as rapid and repeated switching.
 - o Applications must handle system events correctly and comply with Task List behavior guidelines.
- Capability Related Tests (CAP) which emphasize application behavior relating to specific Capabilities, for example:
 - o Multimedia applications must not interfere with voice calls.
 - o Applications which access the mobile network must have network operator approval.
 - o Phone applications must present a UI enabling end user control of the application.
 - o VoIP applications must present a Device Manufacturer disclaimer, and must not interfere with GSM-based telephony functions, including the ability to make emergency calls.

Tips for Testing

The following tips are provided to help you anticipate the tests your application is expected to pass.

The Symbian Signed Wiki pages (developer.symbian.com/wiki/display/pub/Symbian+Signed) also contain useful information, comments and feedback on topics related to testing, including best practice advice, notes on common causes of test failure, currently available tools and utilities, and details of security and signing related error messages developers may encounter when deploying to current UIQ and S60 devices.

Get the Basics Right

Don't fail testing for trivial reasons:

1. Make sure the application user guide is up to date and matches the actual application behavior, and make sure the application version number matches that specified in the SIS file and in the application's 'About' dialog (using major, minor, build number format).
2. Make sure all your UIDs are correct (refer to the Test Criteria for more details):
 - Only use unprotected application UIDs in the range 0xA0000000 - 0xFFFFFFFF for applications you do not intend to sign.
 - Use protected application UIDs in the range 0x20000000 to 0x2FFFFFFF for Express and *Certified Signed* v9.x applications.
 - Only use development UIDs in the range 0x00000000 – 0x0FFFFFFF for test purposes, not for distribution.
 - Specify the correct target platform UID in the application PKG file.
 - Use the correct vendor UID if you have one, or default to 0 otherwise.
3. Follow the file naming recommendations to avoid name clashes with DLLs belonging to other applications, naming your binaries in the format `MyBinaryName_UID.dll` or `MyApplicationName_UID.exe`.
4. If your application is being included in a device firmware build, supply both a normal SIS file and a stub SIS file for signing to ensure that the certificates match.

When testing, you will need at least one suitable phone of the type that you are targeting. You should assume that any personal data on the phone risks being lost, and ideally you should use a dedicated test phone which has been restored to factory settings.

You will also need at least the following:

- a secondary phone with an ability to call and SMS/MMS the phone under test
- any external media which may be required by the application
- media for mass memory devices
- PC connectivity suite and appropriate connection hardware, or any other mechanism to install the application.

Understand the Test Cases

The Symbian Signed Test Criteria describes in detail the tests that your application is expected to pass. As the following summary indicates, most of the tests are aimed at ensuring that applications achieve basic standards of robustness and reliability; a smaller number of tests are based on specific platform features and requirements for applications that include VoIP functionality:

1. Test your application through the install/uninstall/reinstall cycle, including installation from and to removable media, to ensure that install is correct (for example, files are installed to correct locations and resources install correctly), uninstall is clean (no install files are left behind), and that reinstall succeeds.
TESTS UNI-01, UNI-08, UNI-09
2. Profile or otherwise debug/test your application through the start/quit/restart cycle,

including testing under low memory and low storage conditions, testing for startup time, clean shutdown and clean restart, correct response to the system shutdown message, and memory leaks.

TESTS UNI-01, UNI-03, UNI-04, UNI-06

3. Test the running application under conditions of stress including low memory and/or storage, rapid switching to/from the application, bad and boundary case inputs, large file loading, memory card and power loss, network and cable/connection loss, low memory and loss of memory (for example, when all free memory is dynamically consumed by another application).
UNI-02
4. Test the running application for ‘good citizenship’ when other applications install, start, and run, and with interrupting system events (for example, alarms), hardware events (for example, power cable insertion/removal), and telephony services (incoming calls and messages). In particular applications with `MultimediaDD` Capability must not interfere with voice calls and should pause audio when voice calls are incoming.
UNI-02, UNI-05, CAP-02
5. Test your application’s backup and restore functionality.
UNI-07
6. Ensure the running application is visible to/responds to the Task List/Task Manager.
UNI-05
7. Support UI flipping and scalable UI modes and behavior.
UNI-02, UNI-10
8. Test autostart behavior; applications should not automatically restart by default when the phone is rebooted, and autostart settings must be fully under the user’s control.
UNI-11
9. Phone/VoIP applications must comply with specific conditions specified in the Test Criteria.
CAP-03, CAP-04, CAP-05, CAP-06

Make Continuous Testing Part of Your Methodology

Testing is often treated as a secondary activity compared with software design and writing code. The best lesson of recent agile methodologies is that it need not be that way; some agile practices begin with writing test code before a line of production code is written.

A number of test tools and frameworks are available that are customized for Symbian OS, and that can help improve your testing, for example, by providing ready-made unit test frameworks.

A list of useful tools for Symbian Signed can be found at the Symbian Developer Network Wiki, follow the **Wiki > Symbian Signed > Tools** for Symbian Signed tabs from the Wiki home page, developer.symbian.com/wiki.

UIDs for Symbian Development

What are UIDs? What's the difference between “Protected” and “Unprotected range” UIDs?

Any UID values less than or equal to 0x7FFFFFFF are classed as "protected" and are only intended for use with signed applications (or those pre-installed in ROM). The software installer will refuse to install an unsigned application if it uses a package UID in the protected range. New UID allocations (shown below) will start from 0x20000000 for the protected range, and from 0xA0000000 for the unprotected range.

UID	Class Range	Purpose
Protected Range		
0	0x00000000 - 0x0FFFFFFF	Development use only
1	0x10000000 - 0x1FFFFFFF	Legacy UID allocations
2	0x20000000 - 0x2FFFFFFF	V9 protected UID allocations
3	0x30000000 - 0x3FFFFFFF	Reserved
4	0x40000000 - 0x4FFFFFFF	Reserved
5	0x50000000 - 0x5FFFFFFF	Reserved
6	0x60000000 - 0x6FFFFFFF	Reserved
7	0x70000000 - 0x7FFFFFFF	Vendor IDs

Unprotected Range		
8	0x80000000 - 0x8FFFFFFF	Reserved
9	0x90000000 - 0x9FFFFFFF	Reserved
A	0xA0000000 - 0xAFFFFFFF	V9 unprotected UID allocations
B	0xB0000000 - 0xBFFFFFFF	Reserved
C	0xC0000000 - 0xCFFFFFFF	Reserved
D	0xD0000000 - 0xDFFFFFFF	Reserved
E	0xE0000000 - 0xEFFFFFFF	Development use only
F	0xF0000000 - 0xFFFFFFFF	Legacy UID compatibility range

From which range should I use a UID for pre and post Symbian OS v9 applications?

Use UIDs for your application from the protected and unprotected range as specified below:

Application	Pre Symbian OS v9	Post Symbian OS v9
Signed	Protected	Protected
Unsigned	Protected	Unprotected

All applications for Symbian OS v5, v6, v7 and v8 must use protected UIDs.

All applications that are or may be Symbian Signed for Symbian OS v9x must use protected UIDs.

Due to UID range restrictions in pre-v9 Symbian OS, use protected UIDs for all signed or unsigned pre-v9 applications. This includes UIDs in the binaries and the accompanying `.application .pkg` file (i.e., SISUID). The `Makesis.exe` tool will throw an error if unprotected UIDs are used in pre-v9 SIS files, and the application might fail to execute when installed.

Note that Symbian OS v9 applications must use allocated and protected UIDs; otherwise the SIS file submission will result in a failure.

What UID range should I use for SDK example or test code in Symbian OS v9?

For Symbian OS pre-v9 use UIDs from the test range 0x01000000 - 0x0FFFFFFF.

For Symbian OS v9, there are two options for UIDs for SDK examples (assuming that they are to be chosen from the unprotected range, so that developers can run the application without a DevCert):

1. Use UIDs from the unprotected range 0xAxxxxxxx by officially allocating UIDs out of your Symbian Signed portal account.
2. Use UIDs from the test range 0xExxxxxxx by choosing non-clashing "random" values from within this range. Note that this replaces the pre-v9.x test range of 0x01000000 - 0x0FFFFFFF (so anywhere you previously used those UIDs you can now use these for Symbian OS v9).

Within the broad area of example/test applications:

Option 1 is probably more suited to projects (e.g., file-manager or games), and option 2 is more suited to programs purely for illustrative/learning/test purposes (e.g., HelloWorld) which will not be redistributed via a SIS file. You may opt for the first option for all examples though, if you wish.

Note that whichever UID range you use, it needs to be stressed that an ISV application should not copy the UIDs from the examples, but should instead allocate their own UIDs from www.symbiansigned.com to avoid any issues.

How do I get new UIDs?

You need to be a registered user on Symbian Signed, go to www.symbiansigned.com to register. Once you have registered and logged in, click on **UIDs** and then **Request** on the left navigation bar. If you are developing an application for pre Symbian OS v9 you can get UIDs from the protected or unprotected range. If you are developing your application for Symbian OS v9 to be signed, you should use UIDs in the protected range before submitting your application to Symbian Signed. If you are not intending to get your application signed, use UIDs in the unprotected range. Unsigned applications with UIDs in the protected range will not install.

What are SIDs?

A Secure ID (SID) is a special use for a UID. In Symbian OS v9 each executable has a SID which, unless explicitly specified by the SECUREID keyword in the application's MMP file, will default to be the same value as the application's UID3. The SID value is not relevant for DLLs as the SID of a process will always be that of its EXE.

A server can permit or reject a call to a particular API based on the client's SID. It also determines the name of the application's private directory.

To avoid confusion, it is recommended that a SECUREID should not be specified in the application's MMP file; instead UID3 should always be specified.

What is a UID3?

The first 12 bytes of any Symbian OS file are used to store three 32 bit numbers (UID1, UID2 and UID3) that identify what type of file it is. UID3 is the number you specify in your MMP file with the keyword UID to uniquely identify your application.

What are VIDs?

A Vendor ID (VID) is another special use for a UID in Symbian OS v9. Symbian has set aside part of the 32 bit UID range (0x70000000 to 0x7FFFFFFF) for VIDs. A VID can be used as a runtime mechanism to check that a binary comes from a particular source. It is specified by using the VENDORID keyword in a project's MMP file; if this is not found the VID will default to zero. The VID of a DLL is not relevant - similarly to the SID, the VID of a process will always be that of its EXE.

Most developers will not have a VID allocated to them and hence will use the default value of 0; VIDs are most useful to network operators and phone manufacturers - for instance, a network operator could use the VID mechanism to allow only applications with a certain VID to access a particular Network API or service.

If you require a VID please post your request and justification as to why you require a VID on the Symbian Signed Forum which can be found at developer.symbian.com/forum/ann.jsps?annID=27.

Does Symbian Signed check that a UID belongs to a developer pre Symbian OS v9.x?

There is no change to the existing Symbian Signed process for pre Symbian OS v9 applications. Developers do not require UIDs from the new system to get their applications signed.

How does Symbian test that a UID belongs to a developer?

When a developer submits their Symbian OS v9 application, during the application upload process Symbian scans the SIS file (note that Symbian Signed only checks the UID ownership in Symbian OS v9 onwards). The Distinguished Name contained within the Publisher ID and the UIDs from within the application are then recorded by the system. The user can find the results of the SIS file scan by following the link from the application information page. The system looks up the UIDs that have been found in the application and displays the owner of the UID that is listed in the UID allocation database next to each UID. It is then easy for the Test House to compare the owner of the UID with the Distinguished Name taken from the Publisher ID. Note: only non-zero VIDs are displayed. If a UID from the file cannot be found in the database then an error will be reported by the system. A Test House would fail this application.

How can I find out which UIDs have been allocated to me?

Once you are logged in to www.symbiansigned.com, select **UIDs** and then **My UIDs** on the left navigation bar. This page lists all the fields from the UIDs database that are associated with your Symbian Signed account. The records are clearly grouped according to whether the UIDs allocated were in the protected or unprotected range, and they are shown alongside the Distinguished Name and Organization Name. Paging is used to limit the number of records displayed at one time with a search facility (including wildcards) on UID, Organization Name and Distinguished Name.

Frequently Asked Questions

The following are some of the most frequent queries about Symbian Signed.

Q. What are Capabilities?

A. A Capability is a token that allows an application to access a protected resource. The Symbian OS Platform Security model defines 20 Capabilities. On Symbian OS, signing is the mechanism used to grant Capabilities to applications; via the signing process, developers request the Capabilities their application needs and, if granted, those Capabilities are then encoded into the digital certificate used to sign the application.

Symbian OS Capabilities are classified as either User or System Capabilities:

- User Capabilities are designed to be meaningful to mobile phone users and, for example, allow applications to send messages or access user data.
- System Capabilities protect system services, device settings, and hardware features.
- Restricted Capabilities are not available via all signing options.
- The most sensitive System Capabilities are referred to as Device Manufacturer Capabilities.

To access the Device Manufacturer Capabilities, you must complete a Capability Request Form and follow the instructions on the Device Manufacturer Capabilities Wizard.

More information on the process for Nokia is available here at www.forum.nokia.com/main/technical_services/testing/index.html.

Q. Which APIs are Capability-protected?

A. On Symbian OS, Capabilities are used to protect APIs that allow sensitive operations. Examples include operations that may access end users' private data; that may create events that are billable; that may interfere with other applications; that may access the mobile phone network; or that may access handset functions that can affect the normal behavior of the phone.

Q. Do all Symbian OS applications have to be signed?

A. From Symbian OS v9, all applications that need to access protected APIs must have the appropriate Capabilities to do so. Approximately 40% of Symbian OS APIs are Capability-protected.

- Applications that use no protected APIs do not need Capabilities, and therefore may not need to be signed, depending on the security policy implemented on the handsets that the application targets. Note that the user will be warned that the software is not trusted and must agree to install the application.

- Applications that use only APIs protected by User Capabilities may not need to be signed, depending on the security policy implemented on the handsets that the application targets. The user will be warned that the software is not trusted and must agree to install the application. The user will also be asked at install time to grant the Capabilities the application requires.
- Applications that use any APIs protected by System Capabilities must be signed.

It is still possible to install unsigned Symbian OS applications on some phones (vendors decide on the security policy for each handset). However, there is no guarantee that Device Manufacturers will continue to allow unsigned applications to install on their devices. For commercial distribution therefore, and for any widescale distribution of freeware or shareware, signing should be considered a requirement.

There is no requirement to sign applications targeted at versions of Symbian OS earlier than v9.

Q. What is a Developer Certificate?

- A. To enable developers to deploy applications for testing or for other limited purposes (including beta testing or field trials), Symbian Signed provides a restricted form of signing based on Developer Certificates. Applications signed with a Developer Certificate are only installable on the device or devices for which the certificate is granted, based on device IMEI. Within its lifetime of 36 months, a Developer Certificate can be used for an unlimited number of signings.

Note that to request and download a Developer Certificate which can be used to sign applications for use on up to 1000 devices, a Publisher ID is required.

Developers without a Publisher ID can submit a SIS file via the Symbian Signed portal for *Open Signed Online* signing only, allowing deployment to a single specified device.

Q. Do all phones based on Symbian OS support signed applications?

- A. All phones based on Symbian OS releases from v9 fully support application signing. Phones based on earlier releases have either limited (v8) or no support for signing (v7 and earlier releases).

Q. How much does it cost to get an application signed?

- A. The costs of the different signing options vary: *Express Signed* costs US\$20 per signing, plus the cost of a Publisher ID (annually \$200); the cost of *Certified Signed* depends on the fee scales of the Test House the developer chooses, typically in the region of 200-500 EUR, plus the cost of a Publisher ID. *Open Signed Online* is free.

Q. Will I have to pay again for each new release of my application?

- A. Yes, each time you sign software you will incur the costs which apply to the signing option you have chosen.

Q. What happens if my application fails testing?

- A. The Test House will provide a test report identifying the reasons for failure. You can track the progress of your application via your Symbian Signed account. For *Certified Signed* you will need to resolve the issues, and resubmit your application for testing. You will have to pay to have your application re-tested, therefore you are advised to test your application thoroughly against the Test Criteria before your initial submission to reduce the likelihood of failure. Re-submitting an application for testing may cost less than the initial submission, but this is test house dependent.

For *Express Signed* the submitter may be required to submit future applications via *Certified Signed* only, for some period or for some number of signings. The failed application may also be revoked.

Q. Does Symbian Signed apply to Java MIDlets?

- A. No, Symbian Signed only applies to software which is distributed in SIS file format; developers of Java MIDlets should sign their applications using Sun's Java Verified scheme.

Signing does apply to software written in other languages, for example, Flash or Python, and is distributed in SIS file format.

Q. I have UUIDs allocated from uid@symbiandevnet.com. Can I still use them?

- A. A requirement for getting a Symbian OS v9 application signed is that the UUID comes from the new system and is in the protected range. Even if you have previously obtained a UUID from Symbian it will be necessary to reapply at www.symbiansigned.com regardless of whether or not you are intending to sign your application.

You can also continue to use your existing allocations for unsigned application usage on Symbian OS v9. To do this, simply replace the first hex digit (a 1) with F, and leave the remaining digits unaltered. This maps your UUID into the Legacy UUID compatibility range where it will not conflict with any other allocations. For example, you have a UUID allocation 0x100F55BE which you can transpose to 0xF00F55BE for use in an unsigned Symbian OS v9 application.

Symbian Signed with Manufacturer Capabilities for Nokia

This section explains how Nokia grants Device Manufacturer Capabilities for Symbian Signed.

The process steps are as follows:

- A user requests a Developer Certificate (*Open Signed Offline*) for AllFiles, DRM, and/or TCB through the Symbian Signed website.
- The request is evaluated from a technical and commercial point of view by Nokia. Nokia may contact you if there is a need for more information to be submitted.
- If the evaluation is successful, a legal agreement related to DRM liabilities must be signed.
- The Developer Certificate will then be granted to you. Nokia will provide you with details of how to access the “Symbian Signed for Nokia” website where you can submit the application to be Symbian Signed.
- The testing will be performed against the Symbian Signed Test Criteria and Nokia's own test criteria.

After the application has passed “Symbian Signed for Nokia” testing it is Symbian Signed and the application will be available to download through your “Symbian Signed for Nokia” account.

The application must define the platform ID in the PKG file so that it can only be installed to the devices from the same platform. It is also possible to limit the installation to a specific Nokia device or devices.

It is highly recommended that the part of the application which requires the above-mentioned Capabilities is packaged into a separate SIS file. This SIS file will have the required Capabilities and you can embed this SIS file in the main application SIS file in application distribution. This avoids going through Symbian Signed for Nokia testing when changes are made to the main application SIS file.

Symbian Signed with Manufacturer Capabilities for Sony Ericsson

This section explains how Sony Ericsson grants Device Manufacturer Capabilities for Symbian Signed.

- Request a Developer Certificate (*Open Signed Offline*) for any of the Capabilities `AllFiles`, `DRM` or `TCB` through the Symbian Signed site.
- The request is evaluated from a technical and commercial point of view by Sony Ericsson. For this request to be granted the developer should have a business contact within Sony Ericsson.
- Apply for Sony Ericsson channel access for Symbian Signed at developer.sonyericsson.com/symbiansigned.
- Submit the application and Capability Request Form. The Capability Request Form can be found at developer.symbian.com/wiki/display/pub/Symbian+Signed+Documents.
- Before submitting the application it must be signed with a Publisher ID. A Capability Request Form must also be submitted with the application since all applications are granted on a per application basis.
- The testing will be performed against the Symbian Signed Test Criteria and Sony Ericsson test criteria.

If the application is targeted for a specific device this must also be specified in the PKG file.

After the application has passed “Symbian Signed for Sony Ericsson” testing it is Symbian Signed, and the application will be available through your ‘Symbian Signed for Sony Ericsson’ account.

Who’s Who in the Symbian Signed Process?

TC TrustCenter – Symbian Signed Certificate Authority

TC TrustCenter is a Certificate Authority for Symbian Signed. This means that they issue Publisher IDs to developers and provide the signing of SIS files against the Symbian B Root Certificate through the Symbian website.

Further information can be found at:

www.trustcenter.de/en/products/tc_publisher_id_for_symbian.htm

MPhasis – Symbian Signed Test House

MPhasis provide Symbian Signed testing for all Capabilities including the following Manufacturer Capabilities:

Nokia	AllFiles, TCB, DRM
Sony Ericsson	AllFiles, TCB, DRM

Standard Symbian Signed Testing is around 185 Euros.

Please contact MPhasis at the following email address with any queries related to testing or testing fees:

Symbian.signed@MPhasis.com

Further information can be found at:

developer.symbian.com/wiki/display/pub/Symbian+Signed+Test+Houses

NSTL – Symbian Signed Test House

NSTL provide Symbian Signed testing for all Capabilities including the following Manufacturer Capabilities:

Nokia	AllFiles, TCB, DRM
-------	--------------------

Standard Symbian Signed Testing from NSTL is 250 Euros.

Please contact NSTL at the following email address with any queries related to testing or testing fees:

Symbian.signed@nstl.com

Further information can be found at:

developer.symbian.com/wiki/display/pub/Symbian+Signed+Test+Houses

Sogeti High Tech – Symbian Signed Test House

Sogeti High Tech is a wholly-owned subsidiary of the Capgemini Group.

Sogeti High Tech provides Symbian Signed testing for all Capabilities including the following Manufacturer Capabilities:

Nokia	AllFiles, TCB, DRM
Sony Ericsson	AllFiles, TCB, DRM

Standard Symbian Testing from Sogeti is 560 Euros (this is a guide price).

Please contact Sogeti High Tech at the following email addresses with any queries related to testing or testing fees:

nokiatesting.fr@sogeti.com

symbian.signed.fr@sogeti.com

uti.testing-house.fr@sogeti.com

Further information can be found at:

developer.symbian.com/wiki/display/pub/Symbian+Signed+Test+Houses

Support for Symbian Signed

There are two forums available for Symbian Signed users to post queries related to the Symbian Signed process.

- **Symbian Signed General**

developer.symbian.com/forum/forum.jspa?forumID=2

This forum is for queries and discussion related to the overall Symbian Signed process including the Test Criteria.

- **Symbian Signed Support Requests**

developer.symbian.com/forum/forum.jspa?forumID=54

This forum is for queries related to account problems, registration problems and other administrative queries.

Glossary

Capability	Symbian OS Platform Security protects sensitive APIs with Capabilities. Applications must have appropriate Capabilities to access protected APIs. The Symbian Signed program is the mechanism by which application developers are able to acquire the Capabilities needed by their applications.
Certificate Authority, or CA	An issuer of digital certificates. As 'trusted third parties', CAs form an important part of the trust hierarchy of the Public Key Infrastructure. The Certificate Authority for the Symbian Signed program is TC TrustCenter.
Certificate Request file, or CSR	File format which encodes the data provided by developers when requesting a secure digital certificate. Generated by the DevCertRequest tool.
Content IDs	Content IDs are issued by CAs and are used by developers to perform the actual signing of digital content.
Developer Certificate, or DevCert	Symbian-issued secure digital certificate restricted to specified IMEIs. Applications signed with a Developer Certificate will only run on devices matching the IMEIs to which the certificate is restricted.
Device Manufacturer Capabilities	Trusted Computing Base and System Capabilities which protect the most sensitive system services.
IMEI	International Mobile Equipment Identity. Every mobile phone has a unique identifier encoded into its hardware.
PKG file	File format which defines the contents of Symbian OS installable application SIS files. Publisher ID Secure digital certificate used to sign and authenticate digital content. Forms part of the Public Key Infrastructure.
Publisher ID	Secure digital certificate used to sign and authenticate digital content. Forms part of the Public Key Infrastructure.
SignSIS	Command line developer tool used to sign Symbian OS installable application SIS files. Available for Microsoft Windows and Linux.

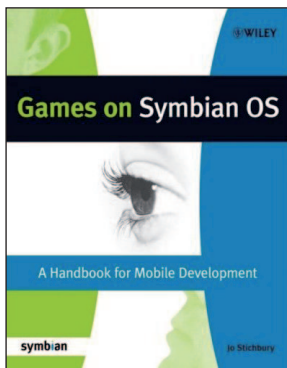
SIS file	File format for installable Symbian OS applications.
Stub SIS file	Stub file generated by the Symbian OS installer providing a condensed application definition.
Symbian Root certificate	The secure digital certificate at the top of the hierarchy of trust defined and implemented by the Symbian Signed program.

New from

symbian
Press

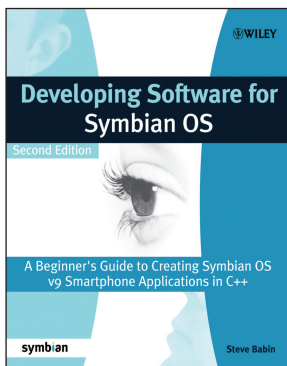


Games on Symbian OS: A Handbook for Mobile Development



This book forms part of the Technology Series from Symbian Press. It describes the key aspects of the mobile games marketplace, with particular emphasis on creating games for smartphones based on Symbian OS v9.x.

Developing Software for Symbian OS, Second Edition



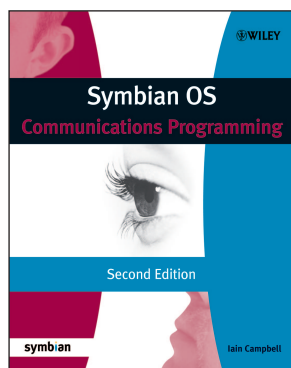
This second edition of *Developing Software for Symbian OS* helps software developers new to Symbian OS to create smartphone applications. The original book has been updated for Symbian OS v9 and now includes a new chapter on application signing and platform security, and updates throughout for Symbian OS v9 and changes to the development environment.

Symbian Press: developer.symbian.com/press

from

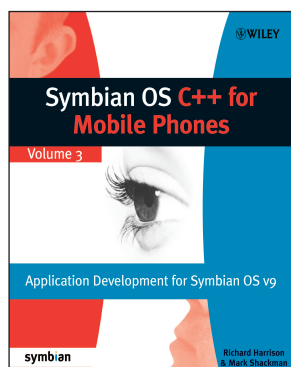
symbian
 Press


Symbian OS Communications Programming, Second Edition



Targeting Symbian OS v9.1 and v9.2, *Symbian OS Communications Programming* - Revised and updated will introduce you to the major communications functionality in Symbian OS and demonstrates how to perform common tasks in each area.

Symbian OS C++ for Mobile Phones, Volume 3



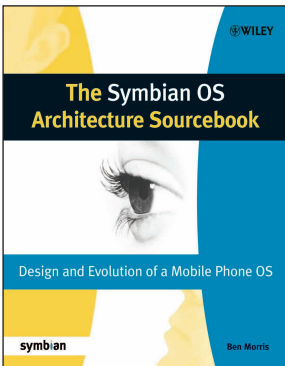
This book will help you to become an effective Symbian OS developer, and will give you a deep understanding of the fundamental principles upon which Symbian OS is based.

New from

symbian
Press

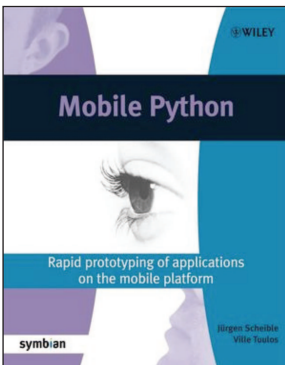


The Symbian OS Architecture Sourcebook



This book conducts a rapid tour of the architecture of Symbian OS and provides an introduction to the key ideas of object orientation (OO) in software, with a detailed exploration of the architecture of Symbian OS.

Mobile Python



Mobile Python is a practical hands-on book that introduces the popular open source programming language Python to the mobile space. It teaches how to program your own powerful - and fun - applications easily on Nokia smartphones based on Symbian OS and the S60 platform.

Symbian Press: developer.symbian.com/press

from

symbian
Press



For all Symbian C++ developers:

Developing Software for Symbian OS
by Steve Babin

Symbian OS C++ for Mobile Phones – Volume 1
by Richard Harrison

Symbian OS C++ for Mobile Phones – Volume 2
by Richard Harrison

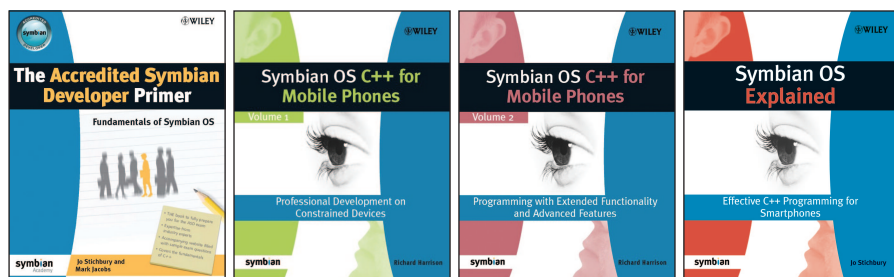
Symbian OS Explained
by Jo Stichbury

Symbian OS Internals
by Jane Sales

Symbian OS Platform Security
by Craig Heath

Smartphone Operating System Concepts with Symbian OS
by Mike Jipping

Accredited Symbian Developer Primer
by Jo Stichbury & Mark Jacobs



from

symbian
Press



For enterprise and IT professionals:

Rapid Mobile Enterprise Development for Symbian OS
by Ewan Spence

For Symbian OS project managers:

Symbian for Software Leaders
by David Wood

For connectivity application developers:

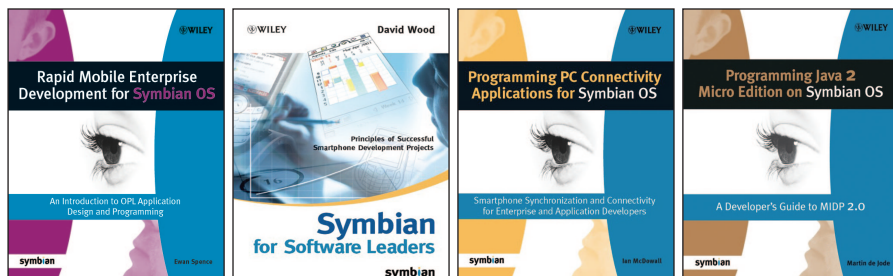
Programming PC Connectivity Applications for Symbian OS
by Ian McDowall

For Java developers:

Programming Java 2 Micro Edition for Symbian OS
by Martin de Jode

For UI Developers

S60 Programming
by Paul Coulton and Reuben Edwards



from
symbian
 Press

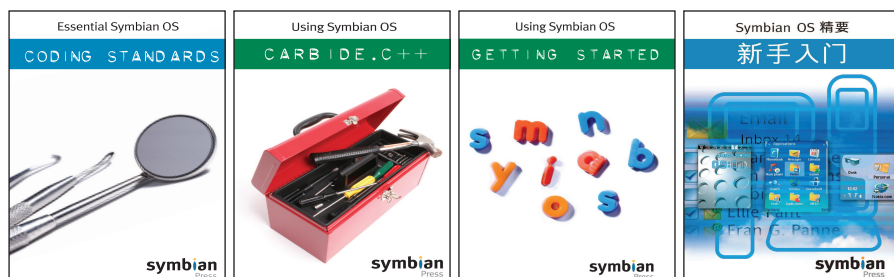


Published Booklets

Coding Standards
 Coding Tips
 Performance Tips
 Essential UIQ - Getting Started
 Getting Started
 Java ME on Symbian OS
 P.I.P.S
 Carbide.c ++
 Data Sharing Tips
 Essential S60 - Developers' Guide

Translated Booklets

Chinese	Spanish
Japanese	Russian
Korean	



developer.symbian.com